CSC 174 Spring 2023
Assignment 4

I do NOT debug for students. Solving assignment problems independently is one assessment criteria of all assignments.

Section 1

Using SQL, create tables according to the given schema in Figure 1. ***You must create your database using exactly the same names for tables and attributes***. The EER is shown in Figure 2.

Views related to the specialization are defined as follows. (please re-type the view definition instead of copy/paste to avoid run time problems)

```
CREATE VIEW TAView As
Select   S.SSN, S.StudentName, S.Address, S.Email, T.Salary
From    Student as S, TA as T
Where  S.SSN=T.SSN;

CREATE VIEW OnlineCourseView As
Select   C.CourseNo, C.CourseName, C.InstructorID, C.NoOfStudents, C.TASSN, W.URL
From    Course as C, OnlineCourse as W
Where   C.CourseNo = W.CourseNo;


CREATE VIEW InPersonCourseView As
Select   C.CourseNo, C.CourseName, C.InstructorID, C.NoOfStudents, C.TASSN,   T.ClassTime,
T.RoomNo, T.Building
From    Course as C, InPersonCourse as T
Where   C.CourseNo = T.CourseNo;
```

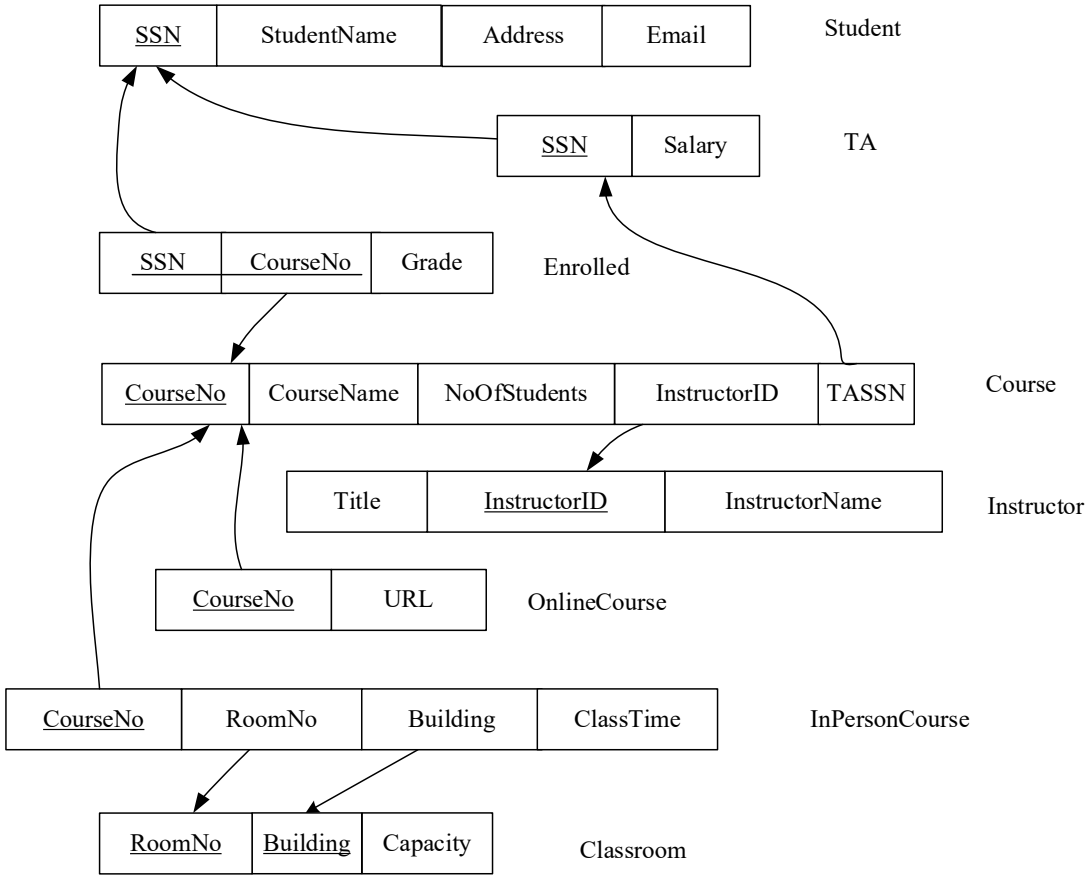

Figure 1. Relational Schema

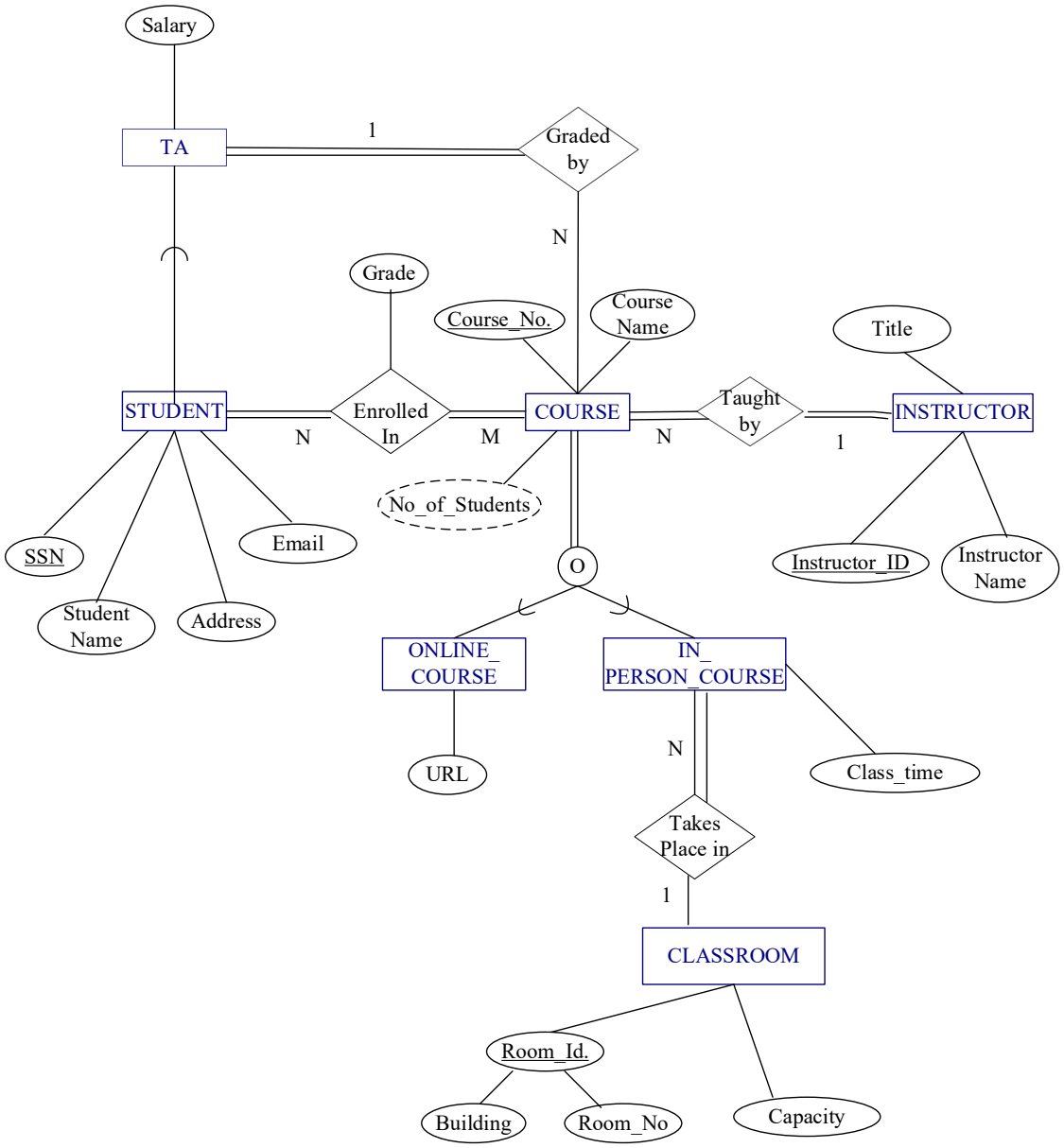

Figure 2. EER

Section 2

Populate the database. See Section 3 for more details.

Section 3

Create the following **views**. Display the results of selecting all tuples from the views. When you populate the database, insert data such that at least one tuple will be display as the result of querying the views (select all tuples from the views).

1) A view constructed by a list of students' names and email addresses who are teaching assistants. Also get the course names for which they are teaching assistants.

   Name of the view: TA_Course

   Attributes of the view: TA name, TA email, Course name

2) A view constructed by a list of students who got at least 2 "A" in their course.

   Name of view: Student_Grade_A

   Attributes of the view: student ssn, number of A's gotten

Section 4

Create a **function** to implement the following requirements:

1) Retrieve the instructor's name who teaches a given course.

   Function name: Course_Instructor

   Input parameter: course name

   Return: Instructor name

When you populate the database, insert data such that the function can return an instructor name.

Section 5

Design stored procedures to implement the following requirements:

1) A stored procedure to output the names of the TAs for a given instructor. The instructor can teach multiple courses, so retrieve the TAs for all his/her courses.

   Name of the procedure: Get_TA

   Input parameter: instructor ID.

   Output: print TAs' names (use select statement, as shown in the examples of MySQL procedure slides)

   2) A stored procedure to output all the students enrolled in a given course.

   Name of the procedure: GetStudentCourse

   Input parameter: Course Number

output: Print SSN, Student Name, Address and Email (use select statement, as shown in the examples of MySQL procedure slides)

When you populate the database, insert data such that at least one result will be display as the result of calling your store procedure.

Section 6

Specify the statements to drop all the tables, views, functions, and procedures. Pay attention to the order of the drop statements in order to drop everything successfully.

Submission

Submit the following files to Canvas. Do NOT zip your files. Zip your files will get a Zero. Please check the syllabus to see what you need to pay attention to when submitting an assignment.

You must execute each statement before submission. **0 point will be given to each non-executable.**

Some items are the same as your Assignment 3, such as create table statements. You can reuse it, but you have to resubmit it again in this assignment. Without the create table statements, I cannot run your procedures nor functions (so you will get a 0 for this assignment).

1. Create table statements (file name must be: 1_create_table.txt)
   I will copy everything from this file using ecs-pd-proj-db **grader account (not your account) to execute it. Make sure the tables are listed in the correct order such that I can execute it without any error. For any table that cannot be created, you will lose points, even if the error caused by the incorrect order of creating table.** Pay attention to the fact that table name and attribute names are case sensitive when using ecs-pd-proj-db server, but may not be case sensitive in your local machine.
2. Insert statements to populate database (file name must be : 2_populate_db.txt)
3. Create view statements (file name must be: 3_view.txt) This file should include all the views given (same as Assignment 3) as well as views in Section 3.

4. Create function statement. The statement to call your function to generate a result. (file name must be: 4_function.txt)
5. Create procedure statements. The statements to call your procedures. (file name must be: 5_proc.txt)
6. Statements to drop functions, procedures, views, and tables (file name must be 6_drop.txt)
7. A PDF file include screenshots (in the following order): (1) call the function and the results displayed. (2) call the procedures and the results displayed.